

REMARKS

In the non-final Office Action mailed January 25, 2008, the Examiner: (i) rejected claim 11 under 35 U.S.C. 112, first paragraph, as failing to have support for “replacing the list of numbers in the clipboard with the summary information”; (ii) rejected claims 11 and 22 under 35 U.S.C. 112, first paragraph, as failing to have support for “in response to the selection event, automatically converting the list of elements to a required type for the desired summary function”; (iii) rejected claim 24 under 35 U.S.C. 112, first paragraph, as failing to have support for “removing the at least one separator from the list of elements”; (iv) rejected claims 11 and 22-24 under 35 U.S.C. 103(a) as unpatentable over US2004/0049729A1 (Penfield) in view of US2002/0124016A1 (Rank et al), 2002/0036662 (Gauthier), and Examiner’s Official Notice. The Examiner also objected to Figure 1A as having improperly labeled elements.

In this Response, Applicant has added amended claims 11, 22, and 24. Applicant notes that some of these changes were previously presented, but not entered because they were improperly presented after final. Applicant has also added claims 25-30 in this Response. Applicant notes that, while these specific combinations of claim elements are ‘new,’ some of the new claims contain language similar to that in previously-withdrawn claims.

Claims 11 and 22-30 will be pending after entry of this Response.

II. Rejections under Section 112

The Examiner rejected claim 11 under 35 U.S.C. 112, first paragraph, as failing to have support for “replacing the list of numbers in the clipboard with the summary information.” To the extent still applicable, Applicant respectfully traverses. Support for this element can be found in the *Abstract* (stating that one embodiment includes “replacing the selected list of numbers with the summary information”); *at pg. 2, lines 19-20* (stating that one embodiment includes “replacing the selected list of numbers with the summary information”); *pg. 6, lines 2-5* (explaining that “[s]ome embodiments may also replace list of elements 129 in the clipboard with a table containing the function name(s) and result(s). These embodiments may be desirable because the statistical information can then be easily “pasted” into another application, such as a word processing program”); *pg. 13, lines 7-8* (claiming “replacing the selected list of elements with the summary information”); *pg. 14, line 11* (claiming “replacing the selected list of numbers

with the summary information”); and in originally-filed claims 7 and 11. Applicant notes that the original claims are part of the disclosure.

The Examiner rejected claims 11 and 22 under 35 U.S.C. 112, first paragraph, as failing to have support for “in response to the selection event, automatically converting the list of elements to a required type for the desired summary function.” In response, Applicant has amended claim 11 to recite “identifying a portion of a web page comprising a plurality of numbers,” and “in response to the selection event, automatically: copying the portion of the web page into a clipboard; . . . [and] extracting the numbers from the web page.” Applicant has also amended claim 21 to recite “required input.” Support for new language can be found at least at *pg. 5, lines 17-21* (explaining that “[t]he operating system 124b then parses the selected text to extract the individual elements of the list 129 at block 206. Part of this process includes looking for blanks, commas, tabs, carriage return commands, or other separators commonly used to separate elements in a list. At block 208, the module 125 will then construct an array containing the elements identified at block 206.”)

The Examiner also rejected claim 24 under Section 112, first paragraph, as lacking support for “removing the at least one separator from the list of elements.” Applicant respectfully traverses. Support for this element at least be can be found at *pg. 5, lines 17-21* (explaining that “[t]he operating system 124b then parses the selected text to extract the individual elements of the list 129 at block 206. Part of this process includes looking for blanks, commas, tabs, carriage return commands, or other separators commonly used to separate elements in a list. At block 208, the module 125 will then construct an array containing the elements identified at block 206.”)

III. Rejections under Section 103

A. The proposed combinations fail to teach or suggest “in response to the selection event, automatically . . . parsing the configuration file to extract the least one desired function [or] extracting the numbers from the web page [or] calculating the at least one desired summary function using the extracted numbers to generate summary information [or] displaying the summary information in a pop-up window [or] replacing the portion of the web page with the summary information.

A combination of references can only obviate an invention if the suggested combination teaches or suggests all of the claimed limitations. *MPEP § 2142*. It follows logically that, if

none of the references teach or suggest a particular limitation, the claimed inventions can neither be anticipated nor obviated by any combination of the cited art.

1. Penfield

Penfield discloses a computer-based statistical analysis program that consists essentially of a standard spreadsheet application with hard-coded region in which results are displayed (as opposed to the more common technique of allowing the user to select the cell(s) in which they want to display the results). *Penfield, Fig. 1; ¶¶ [0010] and [0020]*. Like other standard spreadsheets, Penfield includes a GUI in which the user can select a mathematical function and a range of cells over which that function should be calculated. *Penfield, ¶ [0037]*.

The present invention, in contrast, is directed to a different problem. As described in Applicant's background section, there are many situations where computer users want to know the statistical properties of the list, such as its sum, average, and standard deviation. The conventional way to determine this information is to first transfer the list into a spreadsheet application, then program the spreadsheet application to calculate the desired properties. One problem with this method is that the user must have access to a spreadsheet application and know how to program it. Another problem with this method is that the transfer and programming steps frequently take a considerable amount of time and effort from the user. *Background, pg. 1, lines 14-22*.

The trend toward pervasive devices, such as cellular telephones and personal digital assistants ("PDAs") has exasperated these problems. Most pervasive devices have limited memory and processing power, and thus cannot run a full-featured spreadsheet program. Moreover, even if the pervasive devices did have the capacity to execute a spreadsheet program, the rudimentary data entry capabilities of pervasive devices multiply the time and effort required from the user to transfer the data and program the spreadsheet. *Id. at pg. 2, lines 1-6*.

The present invention overcomes these longstanding limitations by allowing the user to simply highlight a list of numbers in text-based application, such as a web browser or word processor. The user then indicates that they want summary information by, for example, selecting the appropriate command from a context sensitive menu or hovering a mouse pointer over the selected list for a predetermined length of time. In response to this indication, the

present invention automatically copies the list of elements into a temporary memory structure, converts the list into the required input type, computes a pre-selected group of statistics about the elements, and then displays the computed statistics in the results window. *E.g., Detailed Description, pg. 5, lines 5-13.*

With these differences in mind, Applicant has previously amended the claims to more clearly distinguish the conventional ‘cut-and-paste-into-a-spreadsheet’ method contemplated in Penfield. Thus, with respect to claim 1, Applicant respectfully submits that Penfield fails to teach “parsing a configuration file to extract at least one desired summary function.” Instead, users define their desired functions using standard spreadsheet GUI techniques. *Penfield*, ¶ [0037]. Nor does Penfield teach “receiving notification of a selection event [and] . . . in response to the selection event, automatically” performing the specific functions in claims 11 and 22. Penfield does not automatically do anything in response to a user selection event, much less the specific “copying,” “parsing,” “extracting,” “calculating,” “displaying,” and “replacing” acts listed in claim 11 or the specific “converting,” “calculating,” and “displaying” acts in claim 22. Nor does Penfield teach “extracting the numbers from the web page” or “converting the list of elements to a required input for the desired summary function.” Penfield specifically states that the user is responsible for ensuring that the selected elements are of the right input type. *Id.*

2. Rank

Like Penfield, Rank is directed at a traditional spreadsheet application, specifically, a file format used to storing spreadsheet data. *Rank, Abstract*; ¶ [0022]. As such, Rank is essentially silent about techniques for displaying information. For at least this reason, Applicant submits that Rank also fails to teach or suggest “parsing a configuration file to extract at least one desired summary function,” and “in response to the selection event, automatically” performing any of the specific acts identified in claims 11 and 22.

3. Gauthier

Gauthier is also a traditional spreadsheet application that allows users to create a refreshable web query. *Gauthier, Abstract*. Gauthier teaches three different methods for establishing the necessary links: copy-paste web query, export to Microsoft Excel, and edit with

Microsoft Excel. *Gauthier, [0096]-[0098]*. The Examiner's identified paragraphs generally reference the third method. *Office Action mailed January 25, 2008 at pg. 7*. In that method, a user can open the web page in a web browser program and then choose an "Edit with Microsoft Excel" option from a drop down menu on the Toolbar. If the Web page was created with the MICROSOFT EXCEL 2002 spreadsheet program (i.e., containing special codes so that it is already in the right input format for the target application), the page is opened using the MICROSOFT EXCEL 2002 spreadsheet program. *Gauthier, [0098]*.

If, however, the web page was created with another program (i.e., not already in the input format required by MICROSOFT EXCEL 2002), the user must use another method, presumably the first method. *Gauthier, [0160]*. In the 'copy-paste' method, the user must first specifically open the web page inside a special window in MICROSOFT EXCEL 2002. The user then manually identifies the content, then cut-and-pastes it into the spreadsheet. *Gauthier, [0109]*. Next, the user must select the desired format for the information, such as 'text' or 'html,' and the desired formatting, such as 'keep source formatting or match destination formatting.' *Gauthier, [0109]-[0110]*.

With these differences in mind, Applicant respectfully asserts that Gauthier fails to teach or suggest "in response to the selection event, automatically . . . extracting the numbers from the web page" or "in response to the selection event, automatically: converting the list of elements to a required input for the desired summary function." Either the information is already coded for MICROSOFT EXCEL 2002, or the user must specify whether it is in text or html format. Moreover, regardless of whether the user selects text or html, Gauthier does not "automatically . . . calculate[e] the at least one desired summary function using the extracted numbers to generate summary information" or "automatically . . . calculate[e] the at least one desired summary function using the converted elements to generate summary information." Neither text nor html format is suitable for calculation. In fact, Gauthier does not teach or suggest "automatically" doing anything with the imported information, much less "automatically computing" the claimed "summary information." Finally, Applicant is unable to locate anything in Gauthier that resembles "automatically . . . displaying the summary information in a pop-up window" or "in response to the selection event, automatically . . . replacing the portion of the web page in the clipboard with the summary information."

These failings are unsurprising; Gauthier is not directed at making it easier to get data into the spreadsheet *per se*. Instead, it is directed at making it easier to dynamically reference another data source. *Gauthier, [0008]*. Thus, for the type of ‘quick and easy’ situations contemplated by the present invention, Gauthier does not provide any advantages over the method described in Applicant’s Background section at pg. 1, lines 14-21.

4. Official Notice

The Examiner took asserts that “it is notoriously well known, in the art, at the time of the invention to replace the selected list of numbers in the clipboard with the summary information.” *Office Action mailed January 25, 2008 at pg. 5*. The Examiner than goes on to describe one such situation, namely that it was “well known in the art that clipboard data is replaced with one set of copied data with another set of copied data.” *Id. at pg. 6*.

Applicant respectfully submits that the traditional cut-and-paste process does not match the claim language. More specifically, claim 11 requires “in response to the selection event [of a plurality of numbers in a web browser], automatically . . . replacing the portion of the web page in the clipboard with the summary information.” In the traditional cut-and-paste process, copying the something from the browser into the clipboard would replace whatever was previously saved in the clipboard (i.e., not the selected portion of the page) with the selected portion (i.e., not the summary information). Nothing else would happen ‘automatically.’

B. Even if the prior art taught or suggested all of the claim elements, there would be no motivation to make the proposed combination.

In the realm of user interface, it is often the case that a useful, new and unobvious invention does not provide the user with any new capability to perform some action which could not previously be performed by other means, but instead, provides the user with the capability to perform the action in a manner which is more efficient, more natural, easier to learn, easier to implement and/or in some other respect, better, from the user interface perspective, than prior art techniques. This distinction is a subtle but important one. It may be observed, for example, that the ubiquity of so-called “personal computers” is due in large part to the fact that graphical user interfaces have made use of such systems comfortable to the average person, who lacks skilled

training as a typist or computer operator. However, in general such GUI's do not provide the user with any new capability which did not previously exist. Almost all system tasks invoked using a GUI interface can also be invoked using older text-based interfaces.

Applicant's invention addresses a specific area of user interface, namely, generating a pop-up summary of selected values in a way particularly suitable for use by a novice user on a pervasive device. As described in Applicant's background section, the conventional way to determine this information was to first transfer the list into a spreadsheet application, then program the spreadsheet application to calculate the desired properties. One problem with this method is that the user must have access to a spreadsheet application and know how to program it. Unfortunately, most pervasive devices, such as cellular telephones and personal digital assistants, have sharply limited memory and processing power, and thus cannot have a full-featured spreadsheet program. Moreover, even if the pervasive devices did have the capacity to execute a full-featured spreadsheet program, the rudimentary data entry capabilities of pervasive devices multiply the time and effort required from the user to transfer the data and program the spreadsheet. *Applicant's Background, pg. 1, line 13 – pg. 2, line 4.*

Applicant notes that *all* of the references relied upon by the Examiner describe features built on top of the conventional, full-featured spreadsheet applications described in Applicant's Background section. As such, the combinations proposed by the Examiner are simply antithetical to purpose of the present invention, namely to provide a way to quickly and easily obtain statistical information about a list of elements without having to launch and program a full-featured spreadsheet application. *Id. at pg. 2, lines 7-8.* For this reason, Applicant respectfully asserts that there would be no motivation to make the suggested combinations.

V. Objections to Drawings

Applicant submits herewith a revised Fig. 1A containing several corrected element numbers and a changed description of element 126b. Applicant has also amended the Specification to correct some element numbers.

IV. Conclusion

In view of the foregoing comments and amendments, the Applicant respectfully submits that all of the pending claims (i.e., claims 11 and 22-30) are in condition for allowance and that the Application should be passed to issue.

Respectfully submitted,

By: 

Grant A. Johnson, Attorney
Registration No.: 42,696
IBM Corporation - Department 917
3605 Highway 52 North
Rochester, Minnesota 55901-7829

Telephone: (507) 253-4660
Fax No.: (507) 253-2382

ROC920030246US1

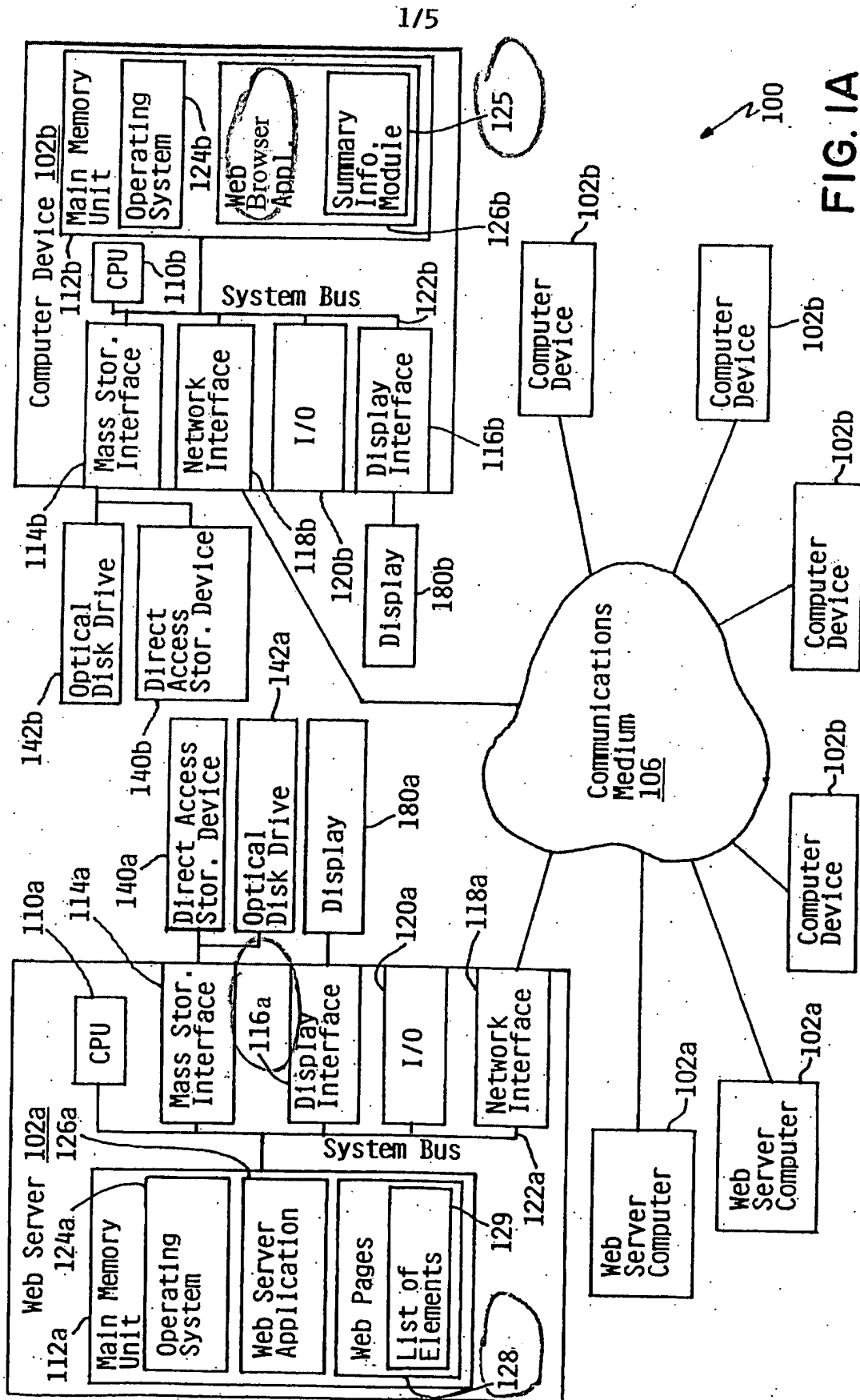


FIG. 1A